Hbase Table Isolation for Multi-tenancy using Region Server Grouping

Motivation:

HBase table isolation is required for scenarios that involve multiple users sharing a common HBase instance. We want to isolate the impact of usage (like read,write patterns) and maintenance activity (like compaction, region splits) of different applications. In the current state, there is no way we can control what regions of a table get assigned to given region servers. A feature to group region servers and assigning tables based on groups will be very useful in multi-tenant environments. In this way, tables belonging to different applications can be assigned to different region server group and hence each application behavior can be isolated to good extent. This design is based on existing work done in the JIRA HBASE-5169. The major difference being usage of command line to manage region server groups and the restriction that a table can only belong to a single region server group at a given time.

Proposal

The work related to this feature can be split into two major components:

1. Developing API to execute commands related to region server grouping.
2. Developing group based assignment manager.

Region Server Group API

To provide commands through hbase shell to do the following:

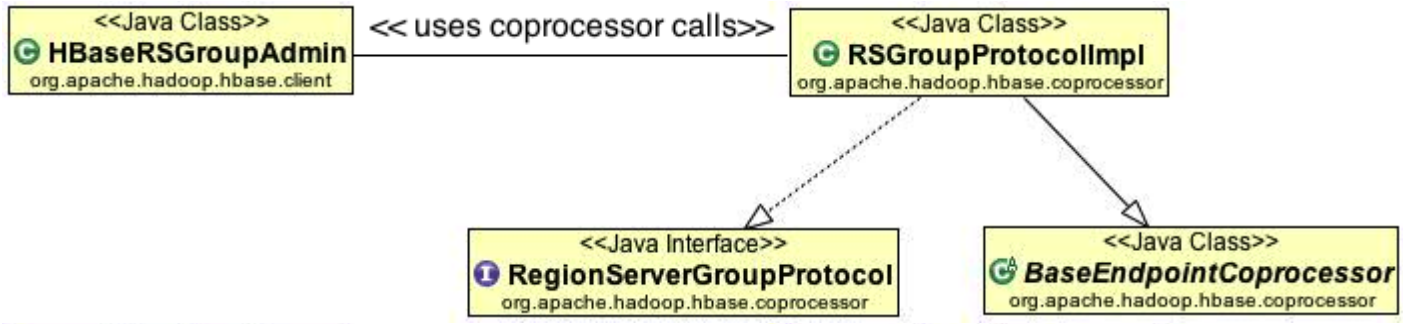
- Create/Delete region server groups.
- Move region servers between groups.
- Move HBase tables between groups.

The following APIs will be required for operations related to grouping region servers and assigning region server groups to tables.

- boolean addGroup(String gName)
- boolean deleteGroup(String gName)
- void moveServerToGroup(String serverName, String gName)

- Map<String, GroupInfo> listGroupMappings()
- Map<ServerName, ServerPlan> listServersInTransition()
- GroupInfo getGroupInfo(String gName)
- GroupInfo getGroupOfTable(String tableName)
- GroupInfo getGroupOfServer(String serverName)
- boolean moveTableToGroup(String tableName, String gName)

We plan to implement all the above functionality using a coprocessor end point.



Similar to the existing hbase commands, the above api's can be called using command line interface. If the HBase instance is configured to not use the region server grouping features, appropriate error messages will be shown to the user.

The following three java classes will be used to manage region server group information:

1. GroupInfo
   This class stores the group information of region server groups. It contains the group name and servers belonging to a group.

2. GroupInfoManager
   This class manages the group information about the region servers.

3. GroupServerManager
   This class extends the existing HBase server manager to add functionality to update the group info manager about the various events happening with the regions.

Group Based Assignment Manager

The group based assignment manager will be an enhancement to the existing assignment manager. The main difference being the current assignment manager assigns regions in random fashion while the group based assignment manager would assign a region only within the group of region servers to which the region belongs. In order to make the new assignment manager pluggable, we will use a factory class to hand over relevant assignment manager to the HBase master.The property "hbase.assignment.manager.class" in the HBase configuration file will be used to determine if the HBase instance will use a default assignment manager or the group based assignment manager.

General Points:

- All the region server group information will be stored in a separate file "**rsgroupinfo.conf**".
- There will be one region group to begin with: **Default.**
- All the other region servers will be assigned to the "Default" group at startup.
- As the groups will be created, region servers will be moved from default group to respective groups.
- Each table can be assigned to only **one** existing region server group.
- All the tables which do not have a region server group assigned will be assigned to the "Default" group.
- The "Default" group cannot be deleted.

The feature to group region servers will be a separate implementation class that will be sub classing the existing assignment manager known as the "RSGroupAssignmentManager". Also, a new implementation of the load balancer class is required, which will be helpful in balancing regions with a group and also balancing regions of a table.

Common Scenarios Walk-Through

Scenario 1 : HBase Instance Startup

The following actions related to region server assignment happen in a sequential manner:

1. HMaster is started.
2. After a server becomes the primary master, it starts to initialize "finishInitialization" method is called. The assignment manager will be initialized to "RSGroupAssignmentManager". The "GroupInfoManager" will look in the hdfs to find if "rsgroupinfo.conf" file is found. If found, it will load the file.
3. As the region servers come up, they are added to respective groups. If no groups are present, then the regions servers are assigned to "Default" group.
4. "ROOT" region assignment takes place. The "assignRoot" method in the assignment manager is used for assignment. If the "ROOT" table has already a group assigned to it , then it is assigned to any of the region servers in the group. If no such group exists, then root region is first assigned to any one of the region servers in the default group.
5. In similar way as the above step, the "META" region is also assigned to a group.
6. Whenever the GroupInfoManager records any new/change in group information, it will immediately reflected in the "rsgroupinfo.conf" file.

Scenario 2: Creating New Region Server Group

1. The Hbase command line shell can be used to manage region server groups. In the shell, issuing "rsaddgroup <group_name>", will invoke the group info manager to create a region server group.
2. Once a region server group is created, region servers can be assigned to it using the command "rsmove <regionserver_name> <group_name>". This command will unassign all the regions assigned to the region server and assign them to the other members of the group. Then the region server will be assigned to the new group. After the move, load balancer will be called on both the old and new groups of the region server.

Scenario 3: Deleting Region Server Group

1. Only empty region server group can be delete.

2. The regions servers belonging to a group can be listed using the command, “rsgroupinfo ‘group_name’”
3. The list of available region server groups can be obtained using the command “rslistgroupmap”
4. Before deleting a group all the region servers in the group need to be moved to different group using the command, rsmove ’regionserver_name’ ‘group_name’.
5. Once the group to be deleted is empty, then the command “rsdeletegroup ‘group_name’” can be used to delete the group.

Scenario 4: New Table Creation

1. In the HBase shell, issue command “create 't1', 'f1', {“RS_GROUP” => 'group_1’}”.
2. If there is no region server group by the name “group_1” , an error will be shown to the user and the create table command will fail.
3. If the create table command is successful, then the table is created and any regions associated with the table will be assigned to region servers belong to the group “group_1”.

HBase Shell Commands

The following command will be added to the existing hbase commands:

1. rsaddgroup ‘group_name’
2. rsdeletegroup ‘group_name’
3. rsmove ’regionserver_name’ ‘group_name’
4. rslistgroupmap
5. rsmovetable ‘table_name’ ‘group_name”
6. rsgroupinfo ‘group_name’
7. rstableinfo ‘table_name’
8. rsinfo ‘regionserver_name’
9. rstransitionlist

References:

- https://issues.apache.org/jira/browse/HBASE-4120
- https://issues.apache.org/jira/browse/HBASE-5169